Journal of Applied Data Sciences
Vol. 6, No. 2, May 2025, pp. 981-996

Nature-based Hyperparameter Tuning of a Multilayer Perceptron Algorithm in Task Classification: A Case Study on Fear of Failure in Entrepreneurship

Theresia Ratih Dewi Saputri[1,*], Edwin Kurniawan[2], Caecillia Citra Lestari[3], Tony Antonio[4]

[1,2,3,4]*School of Information Technology, Universitas Ciputra, CitraLand CBD Boulevard, Surabaya 60219, Indonesia*

(Received: November 15, 2024; Revised: December 18, 2024; Accepted: January 13, 2025; Available online: March 3, 2025)

Abstract

Entrepreneurship plays a key role in generating economic growth, encouraging innovation, and creating job opportunities. Understanding which demographic, psychological, and socio-economic factors contribute to fear of failure in entrepreneurship is essential to developing proper standards in entrepreneurship education and policy. However, it remains challenging to accurately classify these factors, especially when balancing model performance with model complexity in a multilayer perceptron algorithm. An effective model requires the correct parameter setting via a hyperparameter tuning process. Adjusting each hyperparameter by hand requires significant effort and knowledge, as there are frequently multiple combinations to consider. Furthermore, manual tuning is prone to human error and may overlook optimal configurations, resulting in inferior model performance and prediction accuracy. This study evaluates nature-inspired optimization techniques, including particle swarm optimization (PSO), genetic algorithm (GA), and grey wolf optimization (GWO). Several parameters are tuned in the present multilayer perceptron model, including the number of hidden layers and the number of nodes in each hidden layer, learning rate, and activation functions. The used dataset which consists of 39 features from 333 samples captured individual fears, loss score, and computational efficiency as the required amount of time for finding the best parameter combination. Model accuracy performance scores are 45.16%, 53.76%, and 58.61% for GA, PSO, and GWO, respectively. Meanwhile their execution time are 10 minutes, 27 minutes, and 23 minutes, for GA, PSO, and GWO, respectively. Experiment results further reveal that each optimization algorithm has distinct advantages: GA excels at speedy convergence, PSO provides a robust exploration of hyperparameter space, and GWO offers remarkable adaptability to complicated parameter interdependencies. This study provides empirical evidence for the efficacy of nature-inspired hyperparameter modification in improving multilayer perceptron performance for fear of failure categorization tasks.

Keywords: Genetic Algorithm, Particle Swarm Optimization, Grey Wolf Optimization, Machine Learning Model, Artificial Neural Network

1. Introduction

Entrepreneurship plays a key role in generating economic growth, encouraging innovation, and creating job opportunities. Entrepreneurs can lead with technological innovations that can improve production and efficiency across multiple domains by creating innovative product and services. Due to the increased number of new businesses, entrepreneurs have a chance to create new jobs that can help lower unemployment and improve living standards. Even though there many advantages of entrepreneurship, there are still some hesitations to pursue entrepreneurial ventures due to social, economic, and psychological barriers, such as fear of failure (FoF) and perceived risks. This FoF in entrepreneurship is influenced by a combination of personal, socio-economic, and environmental factors. This only adds complexity to the field of study, not only in entrepreneurship and business education but also in psychology [1], [2]. A comprehensive understanding of these factors is crucial for policymakers, educators, and support organizations to create successful strategies that promote entrepreneurship. However, existing approaches to address entrepreneurial FoF are limited by a lack of data-driven methods that are capable of precisely identifying the root causes and predicting individuals' likelihood of participating in entrepreneurial activities [1], [3].

Machine learning models, especially multilayer perceptron (MLP), show great potential in categorizing intricate psychological and socio-economic data. MLPs, a form of artificial neural network, are ideal for tasks involving nonlinear classification and have been effectively used in a range of fields that involve modeling complex relationships

***Corresponding author: Theresia Ratih Dewi Saputri (theresia.ratih@ciputra.ac.id)**
DOI: https://doi.org/10.47738/jads.v6i2.539

in data. MLP has been used in various domains such as medical, economic, and education. In the medical domain, MLP has been employed for medical image classification and segmentation [4] and healthcare chatbots that interact with users to diagnose diseases and suggest treatments [5]. Meanwhile, in education, MLP can be used to predict student performance and identify learning patterns [6] and to analyze the relationships between courses [7]. In economics and business, MLPs are widely used for classification and regression tasks across various fields, including economic forecasting and business analytics [8], [9]. With such complex applications of MLP, it is important to have an effective and efficient MLP models. An effective and efficient MLP model depends on the selection of hyperparameters like hidden layer size, learning rate, and activation functions. It is vital to fine-tune hyperparameters as inadequate parameter combinations can result in less-than-optimal model performance. Unfortunately, manually tuning hyperparameters for MLPs can be error-prone and time-consuming. That is why we need to utilize automated hyperparameter tuning, especially with complex data and parameter combinations.

Our study aims to achieve this by performing a comparative study of automated hyperparameter tuning of MLP for predicting FoF. We perform three different nature-based algorithms: PSO [10], GA [11], and GWO algorithm [12]. Those algorithms were chosen due to their remarkable performance for classification task optimization. Every algorithm has different strengths to find a balance between exploration and exploitation in the search space, making them well-suited for hyperparameter tuning tasks. By systematically comparing these three algorithms, we aim to identify the most effective optimization algorithm for enhancing MLP model performance in predicting FoF based on demographical, psychological, and socio-economic factors. With the help of automated hyperparameter tuning, we are able to capture the optimum combination of parameter for MLP. The optimized MLP model for prediction will help the for policymakers, educators, and support organizations to create successful strategies that promote entrepreneurship.

The paper is organized as follows. Section 2 discusses existing works related to FoF and nature-based optimization for hyperparameter tuning. The research method, including the research questions and proposed approach, is presented in Section 3. Section 4 presents the experiment result and main research. Section 5 provides the conclusion, limitations, and future research opportunities.

2. Literature Review

In this section, we present an overview of the state-of-the-art research related to FoF in Entrepreneurship. We also address the existing work of automated hyperparameter tuning especially focusing on nature-based optimization algorithms.

2.1. Fear of Failure in Entrepreneurship

FoF is defined by [13] as a major psychological obstacle to entrepreneurship that influences an entrepreneur's intention, conduct, and action. FoF might discourage people from pursuing entrepreneurial enterprises since it is generally accompanied with negative emotions like humiliation, embarrassment, and fear over an unclear future. FoF can be felt by all types of individuals, not only among university students and part-time entrepreneurs, but also by business leaders. Muis and Hamid [14] find a negative association between FoF and entrepreneurial inclinations among university students. Specific anxieties, such as humiliation and embarrassment, low self-esteem, and an unclear future, have a substantial impact on entrepreneurial goals. Furthermore, another study found that the humiliation of failing has a detrimental impact on entrepreneurial inclinations. However, social status and entrepreneurial incentive might have a favorable impact on these intents, implying that FoF can be moderated by other motivators [15].

Demographic variables, psychological and emotional issues, social and cultural influences, and economic and career worries are all potential contributors to FoF. According to certain studies, socio-demographic factors such as gender, age, educational level, and work situation can have a substantial impact on perceptions of entrepreneurship [16], [17]. Meanwhile, psychological and emotional aspects such as fear of humiliation and disgrace, concerns about self-worth and personal evaluation, and anxiety about the unknown future can block people from exploring entrepreneurial opportunities [18]. Social stigma can also be a contributing factor to FoF. The societal impression of failure can greatly limit business growth. Additionally, social stigma associated with failure can deter people from taking entrepreneurial risks. Further, the fear of economic and job loss can negatively impact one's decision to become an entrepreneur. This unique anxiety has a detrimental impact on the intention to relaunch a business after a loss, emphasizing the economic risks associated with entrepreneurship [19].

Another important aspect contributing to FoF is financial access. Chapman and Philips in [20] mentioned that a country which have higher economic development to have lower FoF. With the easier financial access, they can provide a safety net that reduce the risk of financial failure that can affect the personal finances of the entrepreneur. Similar to the previous work, research in [21] also mentioned that financial security is one of the major sources of fear with entrepreneurs expressing anxiety about investing their own money into projects.

By integrating this understanding into entrepreneurship education, educators can better prepare students to navigate the psychological challenges of entrepreneurship. The strategy to prepare students navigating the fear can increase the resilient level and innovative entrepreneurial mindset. Educators should find a way to address the socio-cultural factors that contribute to FoF, such as social stigmas and uncertainties, to create a supportive learning environment that encourages risk-taking and innovation.

2.2. Nature-based Optimization for Hyperparameter Tuning

Nature-based optimization algorithms for hyperparameter tuning have attracted significant attention because of their ability to effectively explore complex search areas and increase machine learning model performance. These methods, inspired by natural processes, provide a solid alternative to manual or traditional hyperparameter tuning methods such as grid search. State-of-the-art of nature-based optimization for hyperparameter tuning include GA, PSO, and GWO, each with distinct strengths and uses.

GA uses the genetic principle and the process of natural selection by evolving a population of candidate solutions. The evolution of a population is created with genetic operators, including mutation, crossover, and selection. GAs are useful in scenarios where the search space is large and complex, allowing for a global search that can escape local optimum. Ansari et al. [22] show that GA outperformed the traditional trial-and-error method for hyperparameter tuning. Their experiment demonstrates the effectiveness of GA to identify optimal parameters. Meanwhile, results by Itano et al. [23] further highlight that GA improves the performance of machine learning model.

Another widely known optimization algorithm for hyperparameter tuning is PSO, inspired by the collective behavior of organisms like fish or birds. With this optimization algorithm, the optimal solution is determined by the interaction of particles. The interaction is then evaluated by the fitness function based on its personal best and global best value. [24] proposed a novel approach to combining MLP and PSO. The authors show how this combination can reduce forecasting error in traffic flow prediction. Additionally, [25] shows that PSO give better performance in enhancing MLP performance compared to GA.

Different from GA and PSO, the GWO algorithm is a nature-based algorithm on social dominance hierarchies. The process of optimization starts with a randomized population of grey wolves and is evaluated by a searching for prey process [26] argues that GWO has the ability of handling complex problems in which the dataset has high dimensional and multi-model problems. Another advantage of GWO is that it requires fewer iterations and computational resources to find optimal solutions, making it a cost-effective choice for hyperparameter tuning [27].

While nature-based optimization methods provide major benefits for hyperparameter tuning, they are not without drawbacks. These techniques can be computationally expensive and complex, especially for large-scale issues or in a huge search space. Furthermore, the efficiency of these methods varies depending on the specific problem and dataset, demanding careful selection and calibration of the optimization process. Despite these obstacles, the versatility and efficiency of nature-based approaches make them an applicable method in the area of hyperparameter tuning. Therefore, our work aims to study the performance of each mentioned nature-based optimization methods in term of their effectiveness and efficiency.

3. Methodology

The main objective of this research is to identify the most effective optimization algorithm for enhancing MLP model performance in predicting FoF based on demographical, psychological, and socio-economic factors. To achieve this, we ask:

RQ1: How effective are nature-inspired optimization algorithms for tuning a multilayer perceptron algorithm with the purpose of classification problem of fear of failure in entrepreneurship?

RQ2: What are the advantages and limitation of nature inspired optimization algorithms in terms of Accuracy, Loss Score, and Computational Efficiency?

We address these questions by applying the proposed method shown in figure 1, which adopts a traditional data mining technique [28]. We start with collecting data related to FoF in entrepreneurship. These data are then preprocessed, which includes data cleaning to ensure data consistency, data scaling to change the data into same scale, and feature engineering to determine important features. Next, parameter setup is used to identify the parameter search space that includes hidden layer size, learning rate, and activation function. We also determine the default parameter such as alpha, maximum number of iterations, and random state. Then, we run hyper parameter tuning algorithms: GA, PSO, and GWO. Each algorithm results into a particular model. Each model is evaluated based on four different aspects: accuracy, F1-score, loss score, and computational efficiency.

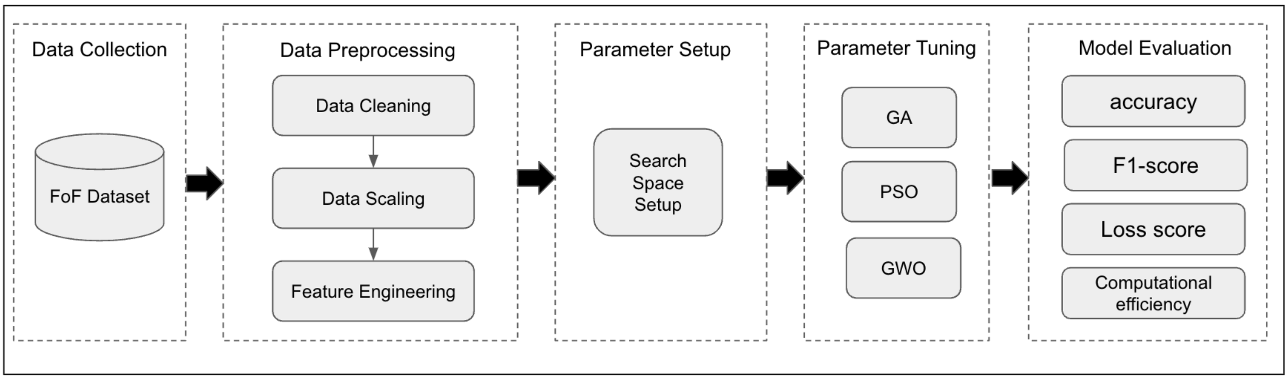

Figure 1. Proposed Method

3.1. Dataset

The dataset used for this search are taken from [29] thanks to its comprehensive attributes that cover demographic information, entrepreneurial intention, and contextual predictors. The dataset consists of 39 features from 333 samples. figure 2 shows five random samples from the raw data. As reported by [29], these self-reported data were collected in February 2021 from there different universities.

Response ID	Age	Gender	Level of Education	University	Entrepreneurial Inclination	EI1	EI2	EI3	EI4	...	Fin1	Fin2	Fin3	Fin4	Fin5
#239	2.0	2.0	2	1.0	1.0	7.0	7.0	7.0	7.0	...	5.0	5.0	5.0	5.0	5.0
#76	1.0	1.0	1	2.0	1.0	7.0	4.0	4.0	4.0	...	4.0	4.0	4.0	4.0	4.0
#199	2.0	1.0	2	3.0	2.0	7.0	7.0	7.0	7.0	...	7.0	5.0	4.0	3.0	5.0
#42	2.0	2.0	2	2.0	2.0	5.0	2.0	2.0	2.0	...	3.0	3.0	3.0	3.0	3.0
#230	3.0	2.0	2	3.0	2.0	7.0	7.0	7.0	7.0	...	3.0	6.0	3.0	3.0	3.0

Figure 2. Raw Data Sample

Demographic information of the dataset include age, gender, level of education, and university. The age feature is formatted into three categories: < 22 years (1), 22- 25 years (2), and above 25 years (3). Gender is divided into two types: male (1) and female (2). Meanwhile, level of education is divided into two responses: undergraduate (1) and postgraduate (2). Entrepreneurial inclination reports the participants' interest towards entrepreneurship. It is numerically coded, which represents varying levels of inclination: low (1), medium (2), and high (3).

The dataset also provides some factors related to entrepreneurial behavior and perception. Those factors consist of seven other entrepreneurial factors: societal support, government support, existing policies, entrepreneurship education, entrepreneurship intention, financial support, and FoF. Each factor is assessed through multiple questions, providing a

ISSN 2723-6471

nuanced view of the influences and barriers related to entrepreneurship. For instance, entrepreneurial inclination may reflect an individual's inherent interest in starting a business, while societal support and government support assess the perceived external encouragement and resources available for entrepreneurship.

3.2. Data Preprocessing

The first process of data preprocessing is data cleaning. This is intended to ensure data are accurate, consistent, and reliable for generating a machine learning model. There are several tasks that can be done in cleaning data such as handling missing values, removing errors, and handling outliers. Using a missing value checker, we find 4 missing values in a column. We handle this by removing the column. We also find inconsistencies in level of education as shown in figure 3. Handling inconsistent data is important in ML task because the data inconsistency can lead to the training bias while generating the model. It can result in poor ML model performance with unreliable prediction result. We addressed this data inconsistency by removing the inconsistent row in the data. We then check the data outlier by visualizing all data using boxplot. Based on our observation, there is no outlier detected in the dataset. Therefore, there is no outlier handling technique applied in this study.

Response ID	Age	Gender	Level of Education	University	Entrepreneurial Inclination	EI1	EI2	EI3	EI4	...
#1	1.0	1.0	1	1.0	2.0	7.0	7.0	7.0	7.0	...
#2	2.0	1.0	1	1.0	2.0	7.0	7.0	7.0	7.0	...
#3	2.0	1.0	2	3.0	2.0	7.0	7.0	7.0	7.0	...
#4	2.0	1.0	2	2.0	2.0	7.0	7.0	7.0	7.0	...
#5	2.0	1.0	2	3.0	1.0	5.0	4.0	7.0	7.0	...
...	...	...	...	...	...	...	...	...	...	...
#329	1.0	2.0	1	1.0	2.0	2.0	2.0	2.0	2.0	...
NaN	NaN	NaN	Chartered Accountant	NaN	NaN	NaN	NaN	NaN	NaN	...
NaN	NaN	NaN	1	NaN	NaN	NaN	NaN	NaN	NaN	...
NaN	NaN	NaN	1	NaN	NaN	NaN	NaN	NaN	NaN	...
NaN	NaN	NaN	1	NaN	NaN	NaN	NaN	NaN	NaN	...

Figure 3. Uncleaned Data Sample

After cleaning all data, the next task is an explanatory data analysis. This task is intended to explore, understand, and summarize the characteristics of the dataset to gain insights, identify patterns, and prepare data for further analysis or modeling. A total of 52% participants are male, and 59% of participants hold an undergraduate degree (see figure 4).

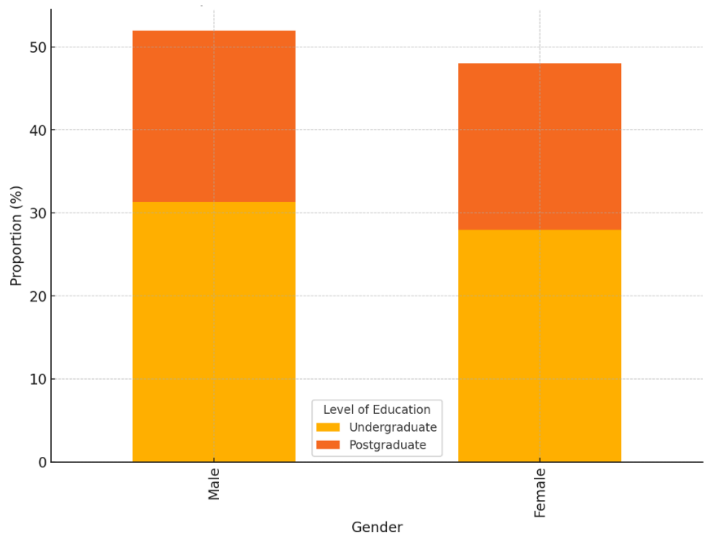

Figure 4. Demographic Distribution of Gender and Education Level

Participants' entrepreneurial inclination is defined as either low or medium interest to start a business (see figure 5). There is no representation for high inclination in the chart, which may imply that none of the participants in this sample are strongly inclined toward entrepreneurship. Notably, medium inclination exhibits a higher frequency when compared to low inclination. These data may indicate that participants are open to entrepreneurial ideas. Given that most participants are at a medium or low inclination level, education institutions might consider introducing educational programs, workshops, or real-world exposure to entrepreneurship to achieve a higher inclination.

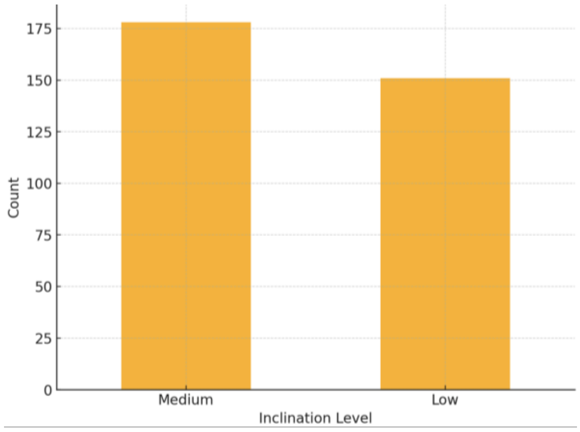

Figure 5. Demographic Distribution of Entrepreneurial Inclination Level

The next task in data preprocessing is data scaling. Since the selected algorithm is MLP, we scale data to a range between 0 and 1 by using the MinMax Scaler in ScikitLearn. Figure 6 shows the result of this scaling method for the first-nine features.

	0	1	2	3	4	5	6	7	8
count	329.000000	329.000000	329.000000	329.000000	329.000000	329.000000	329.000000	329.000000	329.000000
mean	0.287234	0.480243	0.407295	0.490881	0.541033	0.729483	0.691489	0.736575	0.751266
std	0.336347	0.500371	0.492079	0.401873	0.499072	0.309524	0.325323	0.313716	0.314540
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.500000	0.500000	0.500000	0.500000
50%	0.000000	0.000000	0.000000	0.500000	1.000000	0.833333	0.833333	0.833333	0.833333
75%	0.500000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

Figure 6. Excerpt of Data Scaling Results

The last task in data preprocessing is feature engineering. The purpose of feature engineering is to create or modify features (input variables) in a dataset to improve the performance of predictive models. Feature engineering can also help reduce the complexity of models by transforming raw features. This can lead to simpler models that are easier to interpret, can generalize better, and are less prone to overfitting. The feature engineering in this research is summarizing some factors related to entrepreneurial behavior and perception. In this case, the feature creation techniques are selected to get new representative of data. For example, entrepreneurial intention is known to have five indicators. To transform these values, we take the average value of each indicator. We also summarizing the FoF variables which also have five indicators. Finally, we have our preprocessed dataset that consists of 329 sample with 12 features. Those features include Gender, Level of Education, University, Entrepreneurial Inclination, Entrepreneurial Intent, Entrepreneurial Education, Entrepreneurial Motivation, Government Support Policies, Perceived Cultural Support, and Access to Financial Finance.

As seen in figure 7, there is an imbalance dataset for in the data, the over sampling method is implemented in the training set. Imbalance data can cause the bias of machine learning to learn the data patter towards the minor class. Therefore, in this research, oversampling is used over the under sampling due to limited samples. There are various oversampling methods such as Random Oversampling, Adaptive Synthetic (ADASYN), and Synthetic Minority

Oversampling Technique (SMOTE). We implemented the SMOTE oversampling to add more representation of minority class distribution.

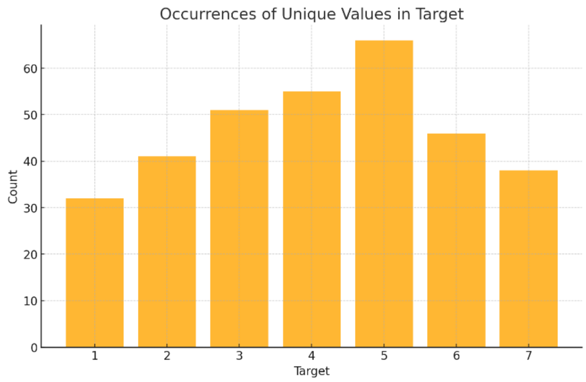

Figure 7. Distribution of Target Variable

3.3. Search Space

Hyperparameter tuning in this research uses the four primary parameters listed in table 1. Each parameter's search option provides a range or list of values explored during the tuning process, often using an optimization algorithm like GA, PSO, and GWO.

Table 1. Hyperparameter Tuning Search Space

Parameter	Search Option
Number of Hidden Layers	1 – 15
Number of Neurons per Layer	8 – 128
Learning Rate	0.001 – 0.05
Activation Function	ReLU, Identity, Logistic, Tanh

The number of hidden layer's search space is defined between 1 and 15 in our research. This range is selected based on the number of feature or input nodes which is 12. According to Stathakis in [30], the selection of number of hidden layers and neuron could be explored using exhaustive and heuristic approach to avoid over fitting. The choice of the number of hidden layers influences the model's capacity to learn complex patterns, with more layers allowing for greater depth but potentially leading to overfitting if not properly tuned. The number of neurons per layer defines the number of neurons within each hidden layer. The search space is set from 8 to 128 neurons. More neurons per layer generally increase the model's ability to capture intricate relationships in the data but also require more computational resources and may risk overfitting on smaller dataset. The learning rate is a crucial parameter that controls the step size of the model's optimization algorithm when updating weights during training. The search range for the learning rate is from 0.001 to 0.05. On the one hand, a smaller learning rate (close to 0.001) allows for more precise adjustments and often results in better convergence but may slow down training. On the other hand, a higher learning rate (closer to 0.05) speeds up training but could lead to an unstable model if set too high. The las parameter specifies the activation function used in the neurons of each layer. The search space includes four options: ReLU (Rectified Linear Unit), Identity, Logistic (usually known as sigmoid function), and Tanh [31].

To maintain the consistency of a generated MLP model, we also set up some default parameter as the control variables. The default parameters are Maximum Iteration (epoch), Alpha, Random State, and Maximum Generation, as can be seen in table 2. Maximum iteration is set to 500 in order to prevent excessive training times, particularly if the model has difficulty converging. If the model achieves convergence before reaching 500 iterations, training will stop early. Alpha is the regularization term set to a value of 0.0001 because we want the model to fit the data closely while still offering some protection against overfitting. Maximum generation, set to 50, specifies the maximum number of generations or cycles that the optimization algorithms. Each generation represents a set of solutions (candidate hyperparameter configurations) that are evaluated and refined to improve model performance. Limiting the maximum generation to 50 helps control the computational cost of the optimization process while allowing enough cycles for convergence toward optimal hyperparameters.

Table 2. Default Parameters

Parameter	Search Option
Maximum Iteration of Multilayer Perceptron	500
Alpha	0.0001
Random State	42
Maximum Generation	50

3.4. Evaluation Metrics

To identify the optimal parameter setting for the model, the effectiveness of each solution generated by the optimization algorithms needs to be evaluated based on accuracy defined by eq. (1), log loss defined by eq. (2), F1 Score defined by eq. (3), and its execution time in minutes. Accuracy measures the proportion of correctly classified instances, thus describing the model’s overall predictive performance. A higher accuracy indicates the model is correctly predicting a large portion of the samples, making it a reliable metric for evaluating model performance in balanced datasets. Log loss evaluates the probability estimates of the model, penalizing incorrect predictions with high confidence more heavily, which helps in understanding the reliability of probability outputs. Log loss evaluates the model’s prediction confidence. Execution time is also considered. Since the data is imbalance, we also measure the F1 Score that are able based on the precision and recall value. This ensures the selected parameter configuration not only achieves high performance but does so efficiently, thereby balancing model accuracy, F1 Score, and computational cost. While accuracy only measure the overall performance, F1 score give better understanding on how the ML model is able to learn the behavior and pattern of each class. Combined, these four metrics offer a comprehensive evaluation framework, enabling the identification of a parameter setting that optimally balances performance and resource efficiency.

$$\text{accuracy} = \frac{\text{number of correct classification}}{\text{total number of sample}} \tag{1}$$

$$\log \text{loss}(y, p) = -(y\log(p) + (1-y)\log(1-p)) \tag{2}$$

$$\text{F1 Score} = \frac{2\times\left(\frac{TP}{TP+FP}\right)\times\left(\frac{TP}{TP+FN}\right)}{\left(\frac{TP}{TP+FP}\right)+\left(\frac{TP}{TP+FN}\right)} \tag{3}$$

Note: TP: True Positive; FP: False Positive; FN: False Negative

3.5. Parameter Tuning

The hyperparameter tuning process in this research is conducted simultaneously with three different optimizations: GA, PSO, and GWO. Each of these algorithms offers unique strategies for exploring the hyperparameter search space, allowing us to find the most effective configurations for our MLP model. For each optimization algorithm, the best solution is evaluated based on two key metrics: accuracy and log loss, with accuracy being the primary criterion for selection. This prioritizing ensures that the chosen solution optimizes the accurate classification rate, which is critical for consistent model performance. When numerous solutions attain the same accuracy, the log loss value is used as a secondary criterion to select the optimal solution. By minimizing log loss, we are able to identify solutions that achieve high accuracy while simultaneously providing solid probability estimates, demonstrating the model's confidence in its predictions. This metric-based priority evaluation procedure aids in identifying the most balanced and effective hyperparameter settings, hence improving model accuracy and resilience.

Figure 8 shows the GA as the first optimization applied for hyperparameter tuning. This algorithm starts with the initialization of a population of potential solutions, each representing a different set of hyperparameters, within predefined bounds. The GA aims to maximize the accuracy score and minimize log loss score. The GA optimization process used in this study involves several key genetic operations: selection, crossover, and mutation.

During the first selection phase, the algorithm chooses the best-performing (i.e., fittest in terms of accuracy and log loss) individuals from the current population. These selected individuals then go through a crossover operation, where portions of their hyperparameter configurations are combined to create new offspring that inherit traits from both parents. This crossover step supports the exploration of new combinations of hyperparameters that may not have been present in the initial population. Mutation is subsequently applied to introduce small, random changes to the offspring's hyperparameters, helping to make sure the diversity in the population and prevent premature convergence to suboptimal solutions.

Once the new generation of solutions is created through selection, crossover, and mutation, each individual is evaluated based on its accuracy and log loss score. Then, the algorithm proceeds to the next generation, using the newly evaluated population as the basis for further evolution. This process of selection, crossover, mutation, and evaluation is repeated until it reaches the predefined maximum generation, allowing the GA to iteratively refine the population and move towards an optimal set of hyperparameters.

Algorithm 1 Genetic Algorithm with Prioritized Metrics and Best Solution Retention

```
Input: Dataset D, Genetic Algorithm parameters, MLP hyperparameter bounds.
Output: Best set of hyperparameters for the MLP classifier.
procedure INITIALIZE POPULATION
    Create an initial population of random MLP hyperparameters within bounds.
end procedure
procedure EVALUATE POPULATION
    for each individual in the population do
        Train and evaluate the MLP on test data to get (accuracy, loss).
    end for
end procedure
procedure TRACK BEST SOLUTION
    Initialize the best solution with default metrics.
    for each individual do
        if individual's metrics are better than the best solution (accuracy ↑, loss ↓) then
            Update the best solution with the individual's metrics and hyperparameters.
        end if
    end for
end procedure
procedure GENETIC ALGORITHM LOOP
    for each generation do
        Selection: Select individuals based on fitness.
        Crossover: Pair selected individuals and apply crossover.
        Mutation: Randomly mutate offspring.
        Evaluate the new generation and track the best solution.
        if generation's best solution is better than the overall best solution then
            Update the overall best solution.
        end if
    end for
end procedure
procedure RETURN BEST HYPERPARAMETERS
    Return the hyperparameters of the best individual.
end procedure
```

Figure 8. Pseudocode of the Genetic Algorithm for Hyperparameter Tuning

Figure 9 shows the second optimization applied for hyperparameter tuning: PSO. Similar to GA, the algorithm starts with the initialization of a particles of potential solutions, each representing a different set of hyperparameters, within predefined bounds. In the beginning of the algorithm, the particle's position and velocity are assigned randomly. The term velocity in PSO refers to the movement direction in searching parameter within the search space.

The same evaluation procedure is used in GA. Once the particles are initialized, each particle's hyperparameters are tested by training and validating the MLP model in the evaluation phase. For each particle, the model's performance is assessed using two prioritized metrics: accuracy and log loss. If a particle's current configuration yields better performance (higher accuracy or lower log loss) than its previously recorded personal best, the particle's personal best is updated to the new configuration. In parallel, the algorithm also maintains a 'global best' solution, which tracks the best-performing hyperparameter configuration found across all particles in the swarm. This global best is updated if any particle's personal best exceeds the current global best metrics, ensuring that the swarm is collectively moving toward the best possible solution.

The main PSO loop then begins, where particles iteratively adjust their positions and velocities to improve their solutions. During each iteration, each particle's velocity is updated based on three components: inertia, which maintains its current trajectory; a cognitive component, which pulls the particle toward its personal best; and a social component, which draws the particle toward the global best solution.

Figure 10 shows the last optimization algorithm, GWO, applied for hyperparameter tuning. Here, the GWO algorithm mimics the hierarchical hunting strategy of grey wolves, where wolves are organized into ranks, and the position updates are guided by the top-performing wolves. The best solution is evaluated using prioritized evaluation metrics (accuracy and log loss) and a best solution retention strategy is used to find optimal hyperparameter configurations. The GWO algorithm starts with the initialization of the wolf population that represents the parameter combination and its position. Each wolf will represent a set of hyperparameter such as the number of layers, neurons per layer, learning rate, and activation function. In this first step, we also define the leaders' rank as alpha for the best-performing wolf (i.e., the optimal solution), beta as the second-best wolf, and delta as the third-best wolf. These leaders guide the movement and behavior of the other wolves, referred to as omega wolves, who follow the leaders to explore the search space. The positions (hyperparameter values) of the wolves are updated based on their relative distances to the alpha, beta, and delta wolves. Using coefficients (A and C) that decrease over iterations, the exploration gradually transitions into exploitation, balancing exploration of the hyperparameter space and convergence toward the best solutions.

Algorithm 2 Particle Swarm Optimization with Prioritized Metrics and Best Solution Retention

Input: Dataset D, PSO parameters, MLP hyperparameter bounds.
Output: Best set of hyperparameters for the MLP classifier.
procedure INITIALIZE PARTICLES
  Create an initial swarm of particles with random positions (hyperparameters) within bounds.
  Initialize particle velocities randomly.
  For each particle, set personal best position to its initial position.
end procedure
procedure EVALUATE PARTICLES
  for each particle in the swarm **do**
    Train and evaluate the MLP using the particle's hyperparameters to obtain metrics:
      - Accuracy (higher is better)
      - Loss (lower is better)
    Update the particle's personal best if current metrics are better.
  end for
end procedure
procedure TRACK GLOBAL BEST SOLUTION
  Initialize global best solution with default metrics.
  for each particle **do**
    if particle's metrics are better than the global best (accuracy $\uparrow$, loss $\downarrow$) **then**
      Update the global best solution with the particle's metrics and hyperparameters.
    end if
  end for
end procedure
procedure PSO LOOP
  for each iteration **do**
    for each particle **do**
      Update particle velocity based on inertia, cognitive, and social components:
        $v \leftarrow w \cdot v + c_1 \cdot r_1 \cdot (p_{\text{best}} - x) + c_2 \cdot r_2 \cdot (g_{\text{best}} - x)$
      Update particle position:
        $x \leftarrow x + v$
      Ensure particle position is within hyperparameter bounds.
    end for
    Evaluate particles and update personal bests.
    Update the global best solution if a better one is found.
  end for
end procedure
procedure RETURN BEST HYPERPARAMETERS
  Return the hyperparameters of the global best particle.
end procedure

Figure 9. Pseudocode of the Particle Swarm Optimization Algorithm for Hyperparameter Tuning

Algorithm 3 Grey Wolf Optimization (GWO) with Prioritized Metrics and Best Solution Retention

Input: Dataset D, GWO parameters, MLP hyperparameter bounds.
Output: Best set of hyperparameters for the MLP classifier.
procedure INITIALIZE PACK
  Create an initial population of wolves (positions represent hyperparameters) within bounds.
  Initialize α, β, and δ wolves based on their initial performance:
    α: best solution, β: second-best solution, δ: third-best solution.
end procedure
procedure EVALUATE WOLVES
  for each wolf in the population **do**
    Train and evaluate the MLP using the wolf's hyperparameters to obtain metrics:
      - Accuracy (higher is better)
      - Loss (lower is better)
  end for
  Set the top three wolves as α, β, and δ based on the prioritized metrics:
    1. Accuracy (higher is better)
    2. Loss (lower is better if accuracies are equal)
end procedure
procedure TRACK GLOBAL BEST SOLUTION
  Track the best overall solution (global best) based on prioritized metrics (accuracy $\uparrow$, loss $\downarrow$).
end procedure
procedure GWO LOOP
  for each iteration **do**
    for each wolf in the pack **do**
      Update the position of each wolf based on α, β, and δ wolves:
        $D_\alpha = |C_1 \cdot \alpha - X|$, $D_\beta = |C_2 \cdot \beta - X|$, $D_\delta = |C_3 \cdot \delta - X|$
        $X_1 = \alpha - A_1 \cdot D_\alpha$
        $X_2 = \beta - A_2 \cdot D_\beta$
        $X_3 = \delta - A_3 \cdot D_\delta$
        Update position: $X_{\text{new}} = \frac{X_1+X_2+X_3}{3}$
      Ensure the new position is within hyperparameter bounds.
    end for
    Evaluate the new positions of the wolves and update α, β, and δ based on prioritized metrics.
    Update the global best solution if a better one is found.
  end for
end procedure
procedure RETURN BEST HYPERPARAMETERS
  Return the hyperparameters of the global best wolf.
end procedure

Figure 10. Pseudocode of the Grey Wolf Optimization Algorithm for Hyperparameter Tuning

The main GWO loop iterates through a defined number of maximum iterations, continuously refining the positions of the wolves and updating their performance metrics. Each wolf's new position is calculated as the average of its distances to the three top-performing wolves, ensuring that the search remains focused on the best solutions discovered thus far. Following each position update, the wolves are re-evaluated, and the ranking is updated based on the wolves' revised performance metrics. Each wolf's hyperparameters are evaluated by training and testing the MLP model. Metrics like accuracy, F1 score (to be maximized) and loss (to be minimized) are used as fitness measures. By the end of the process, the algorithm returns the hyperparameters of the best-performing wolf, providing a solution that maximizes model accuracy and F1 score, and minimizes log loss, thus maintaining the overall effectiveness of the MLP in FoF classification tasks.

4. Results and Discussion

This section first presents the results of the hyperparameter tuning experiment along with the final parameter setting based on the generated best solution in each optimization algorithm. It then discusses how these results answer the proposed research questions.

4.1. Results

Table 3 details the automated hyperparameter tuning experiment results. The baseline MLP model without optimization has a low accuracy of 22.000%, indicating the poor performance of grid search. This model is generated using the default parameter in Sklearn. With optimization, GA significantly improves the model's accuracy to 45.161%, demonstrating the effectiveness of automated tuning. PSO further enhances accuracy to 53.7647%, showing its capability to find even better hyperparameter configurations. GWO achieves the highest accuracy at 58.612%, making it the most effective algorithm in this experiment. The baseline model only achieves 0.160 for the F1-Score indicating the poor model's performance to understand the different class behavior. The GA is able to increase the F1-score to 0.466. PSO is also able to increase the F1-score to 0.539. GWO show the best model performance by increasing the F1-score into 0.574.

In term of log loss, the baseline MLP has a comparatively high 7.548 log loss. This value indicates poor reliability in its predictions. All optimization algorithms significantly reduce log loss, with GA achieving a log loss of 4.354, PSO achieving 3.331, and GWO achieving 3.262 in this study. While GWO achieves the highest accuracy, PSO shows a slight advantage in log loss, indicating that it provides slightly more reliable probability estimates than GWO. The baseline model's execution time is 45 minutes, as it does not involve any optimization. Among the optimized models, GA is the fastest, with an execution time of 10 minutes, followed by GWO at 23 minutes, and PSO at 27 minutes. Although PSO and GWO yield higher accuracy and F1-score, and lower log loss than GA, they require more time to complete the optimization, reflecting a trade-off between performance and computational cost.

Table 3. Experiment Results

Optimization Algorithm		**Evaluation Metrics**		
	Accuracy (%)	**F1-score**	**Log Loss**	**Execution Time (min)**
Grid Search (Baseline)	22.000	0.160	7.548	45
Genetic Algorithm	45.161	0.466	4.354	10
Particle Swarm Optimization	53.764	0.539	3.582	27
Grey Wolf Optimization	58.612	0.574	3.226	23

Table 4 shows the final hyperparameter settings identified by each optimization algorithm. These parameters include the number of hidden layers, the number of neurons per layer, the learning rate, and the activation function. GA selects an architecture with two hidden layers and a varying number of neurons per layer [116, 15], a learning rate of 0.0012, and the Tahn activation function. PSO optimizes the MLP with three hidden layers and neurons per layer [123, 67, 8], using a learning rate of 0.0015 and the ReLu activation function. This configuration yields the high accuracy, F1-score and lowest log loss, indicating that PSO found a balanced solution with fewer layers and specific neuron allocations for each layer. GWO configures the MLP with six hidden layers and neurons per layer [36, 128, 25, 83, 60, 27], a learning rate of 0.0011, and the ReLu activation function.

ISSN 2723-6471

Table 4. Final Parameter Setup

Optimization Algorithm	Parameter Setting			
	Number of Hidden Layer	Neurons per Layer	Learning Rate	Activation Function
Genetic Algorithm	2	[116, 51]	0.0012	Tahn
Particle Swarm Optimization	3	[123, 67, 8]	0.0015	ReLu
Grey Wolf Optimization	6	[36, 128, 25, 83, 60, 27]	0.0011	ReLu

4.2. Discussion

RQ 1: How effective nature inspired optimization algorithms for tuning MLP algorithm with the purpose of classification problem of FoF in entrepreneurship?

Based on the result presented in table 3, we can see the selected optimization algorithms are effective for tuning algorithm in MLP. Even though the accuracy and F1-score of models is small, all of the algorithm outperformed the baseline model in the entire evaluation metrics. The small number of accuracies may be caused by the imbalanced dataset as seen in figure 7 and limited number of samples. Therefore, the further work of this study is to collect more samples from different country and university to ensure the generalization of the dataset.

However, the effectiveness of all algorithms is twice better compared to the baseline model. This result show that nature-based optimization algorithms proven effective in capturing complex patterns within psychological data related to FoF. Traditional tuning methods, for example grid-based tuning, are limited in their ability to explore a broad hyperparameter space, thoroughly. On the other hands, the nature-inspired algorithms in this study provide a dynamic and adaptive search process that is better suited to learn about the complex and multi-dimensional data. By dynamically adjusting hyperparameters and exploring a more extensive search space, GA, PSO, and GWO allowed the MLP to model complex patterns in the social science data related to FoF. In terms of efficiency, the computational costs of the three optimization algorithms are varied based on its execution time.

In terms of their stability and consistency, there are some different performances of the algorithm based on the convergence level. GA has a moderate convergence rate where the procedure converges after seven number of iterations. While GA is generally faster in reaching an adequate solution, unfortunately it may not always achieve the best global optimum because its convergence can sometimes be prematurely driven by the dominance of high-performing solutions in early generations. While GA is stable in its ability to reach a solution quickly, it may exhibit inconsistency in finding optimal or near-optimal solutions due to its tendency to converge on local optimal. PSO typically has a slower convergence rate than GA, as it is designed to thoroughly explore the search space by balancing each particle's movement toward both its personal best and the swarm's global best. In this experiment, PSO required the longest time (27 minutes) to converge, indicating a deliberate and comprehensive search process. It converged after iteration tenth. PSO's consistency is one of its strengths, as its swarm-based approach tends to reach similar high-quality solutions across runs, making it a stable choice for hyperparameter tuning. Although PSO requires more processing time, it demonstrates consistent performance in finding robust solutions, making it suitable for applications where stability and optimal solution quality are essential, even if it means sacrificing some computational efficiency. GWO has a balanced convergence rate, as it combines exploration and exploitation by guiding wolves toward the positions of the top three solutions (alpha, beta, and delta). In this experiment, GWO took 23 minutes, positioning it between GA and PSO in terms of convergence speed. GWO converged after nine iterations. This indicates that GWO can efficiently explore the search space without sacrificing too much time, achieving a convergence rate that is both steady and effective. GWO's balanced of stability, moderate convergence rate, and strong solution quality make it a better choice for tasks requiring dependable results within a moderate timeframe.

Using the MLP model with optimized parameter setup based on GWO result, we then analyzed the factor that contribute to FoF prediction. The importance score for each feature is presented in figure 11. From this result we can see that Financial Support (feature 10) plays significant role to predict FoF. This means individuals' perceptions of available financial resources significantly influence their confidence and willingness to pursue entrepreneurial activities. This could imply that when individuals feel they have adequate financial backing or access to funding sources, they may experience less fear related to the financial risks associated with starting a business. In addition to financial support, Entrepreneurial Intention (feature 6) is also an important predictor of FoF. Those with strong entrepreneurial intentions are likely to have a clearer vision, stronger motivation, and more resilience, which can mitigate FoF and increase their likelihood of pursuing entrepreneurial goals despite the inherent risks. Another significant feature is Entrepreneurial Education (feature 7), which also plays a notable role in predicting FoF. Entrepreneurial education improves individuals' entrepreneurial performance with knowledge, skills, and strategies essential for handling the challenges of entrepreneurship. This educational background can enhance their self-efficacy and preparedness, reducing the uncertainties and fears associated with starting and managing a business.

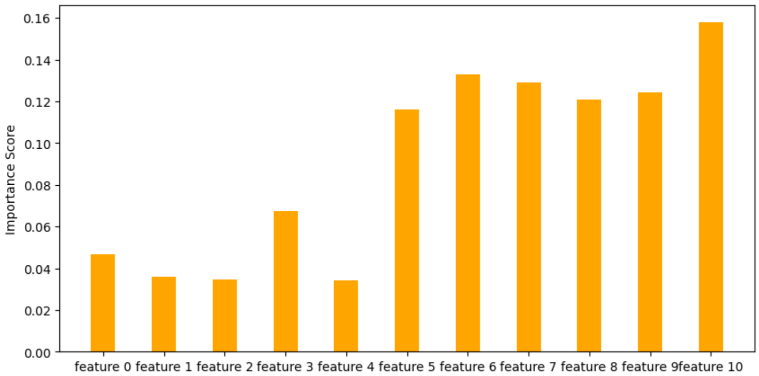

Figure 11. Distribution of Features Importance

RQ 2: What are the advantages and limitation of nature inspired optimization algorithms in terms of Accuracy, F1-Score, Loss Score, and Computational Efficiency?

In terms of efficiency, the computational costs of the three optimization algorithms are varied based on its execution time. GA was the fastest of the three algorithms, taking only 10 minutes to complete the hyperparameter tuning. This lower computational cost can be attributed to GA's efficient selection, crossover, and mutation operations, which iteratively refine solutions without excessive exploration. However, despite its efficiency in terms of processing time, GA achieved a lower accuracy and F1 Score (45.161% and 0.466) compared to PSO and GWO, indicating that while it is computationally efficient, it may not explore the search space as thoroughly. GA's relatively higher log loss (4.354) also suggests that the solutions it found were not as reliable in terms of probabilistic predictions. Therefore, GA may be suitable in scenarios where time efficiency is prioritized over the highest possible model performance.

Meanwhile, PSO required 27 minutes to complete the tuning, making it the most computationally intensive of the three algorithms. This longer processing time is partly due to the iterative adjustments of each particle's velocity and position based on personal and global best solutions, which enhances the algorithm's ability to thoroughly explore the search space. PSO achieved a high accuracy of 53.767% and the lowest log loss at 3.582, indicating that it provided both accurate and reliable probability estimates. Although PSO's computational cost is higher, the improved model performance suggests that it is well-suited for applications where accuracy and reliable probability outputs are critical. PSO's thorough search makes it ideal for complex datasets where capturing intricate patterns outweighs the need for fast processing.

Being in the middle, GWO took 23 minutes to complete the hyperparameter tuning, positioning it between GA and PSO in terms of processing time. GWO's structure, which involves position updates based on the top three wolves (α, β, and δ), allows it to strike a balance between exploration and exploitation, leading to efficient convergence. GWO achieved the highest accuracy (58.612%) among the three algorithms, with a log loss of 3.626, which is slightly higher than PSO but still lower than GA. This indicates that GWO was effective in finding a well-optimized configuration within a moderate time frame, providing a good trade-off between processing time and performance. GWO's relatively high accuracy and F1-score and moderate computational cost make it suitable for applications that require robust model performance without excessive computational resources. Once there is larger dataset, more advance GWO algorithm are needed. The work proposed by Makhadmeh et al. in [32] show some variants of GWO algorithm such as hybrid GWO called min-conflict algorithm with an efficient local search method.

5. Conclusion

Nature-based hyperparameter tuning shows a promising result in achieving a more effective and efficient ML model especially in a MLP model. The nature-based algorithm is able find the best combination of the hyperparameters (number of layers, activation function, number of nodes per layer, and learning rate) given the search space to create MLP model to predict FoF based on demographical, psychological, and socio-economic factors. Adding a broader boundary for the search space and adding more dimension in the search space might help these nature-based algorithms to find a better solution for the hyperparameters.

Overall, PSO and GWO demonstrate greater stability and consistency in capturing complex patterns in the data, with PSO favoring a comprehensive search and GWO providing a middle-ground solution. The choice between these algorithms should depend on the specific needs of the application, whether that's the highest possible accuracy and F1-score (PSO), balanced effectiveness and efficiency (GWO), or faster, lower-cost solutions (GA). Further studies may also include experimenting in another nature-based algorithms (other than GA, GWO, and PSO) to further explore the effectiveness and efficiency of nature-based algorithm automated hyperparameter tuning.

Even though the proposed optimization algorithm for automated parameter tunning outperformed the traditional method, the overall accuracy is still relatively low. There are some possible causes of this issue such as imbalance dataset, limited number of samples, and the variation of dataset. Therefore, the further study should include more advance method for handling data imbalance. The further study also needs to be improved by adding more samples to improve its generalization.

6. Declarations

6.1. Author Contributions

Conceptualization: T.R.D.S., E.K., C.C.L., and T.A.; Methodology: C.C.L.; Software: T.R.D.S. and E.K; Validation: T.R.D.S., C.C.L., and T.A.; Formal Analysis: T.R.D.S., C.C.L., and T.A.; Investigation: T.R.D.S.; Resources: T.R.D.S.; Data Curation: C.C.L.; Writing Original Draft Preparation: T.R.D.S., E.K., C.C.L., and T.A.; Writing Review and Editing: C.C.L., T.R.D.S., and T.A.; Visualization: T.R.D.S. All authors have read and agreed to the published version of the manuscript.

6.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

6.3. Funding

This research has been funded by The Ministry of Higher Education of Indonesia (contract #048/SP2H/PT/LL7/2024).

6.4. Institutional Review Board Statement

Not applicable.

6.5. Informed Consent Statement

Not applicable.

6.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] S. K. Halim, D. Hidayat, Y. Eni, and E. Fernando, "What is Entrepreneurial Fear of Failure?," *Binus Business Review*, vol. 14, no. 1, pp. 73–84, 2023.

[2] E. Hunter, A. Jenkins, and C. Mark-Herbert, “When fear of failure leads to intentions to act entrepreneurially: Insights from threat appraisals and coping efficacy,” *International Small Business Journal*, vol. 39, no. 5, pp. 407–423, 2021.

[3] Y. Gao, X. Wang, J. Lu, B. Chen, and K. Morrin, “Entrepreneurial fear of failure among college students: A scoping review of literature from 2010 to 2023,” *Heliyon*, vol.10, no. 10, pp. 17-29 , 2024.

[4] Zhang, M., Wen, G., Zhong, J., Chen, D., Wang, C., Huang, X., and Zhang, S., “MLP-like model with convolution complex transformation for auxiliary diagnosis through medical images,” *IEEE J Biomed Health Inform*, vol. 27, no. 9, pp. 4385 - 4396, 2023.

[5] M. K. Ogirala, R. Tallapaneni, S. M. Chalamcharla, and A. Chinta, “A Medical Diagnosis and Treatment Recommendation Chatbot using MLP,” in *2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, vol. 2023, no. 06, pp. 6, 2023.

[6] Z. Ersozlu, S. Taheri, and I. Koch, “A review of machine learning methods used for educational data,” *Educ Inf Technol (Dordr)*, vol. 29, no. 05, pp. 1–21, 2024.

[7] J. Shi, T. Wu, Y. Lei, and B. Li, “Course Correlation Analysis using MLP,” in *2023 International Conference on Cyber-Physical Social Intelligence (ICCSI)*, vol. 2023, no. 12, pp. 279–284, 2023.

[8] A. Tashakkori, M. Talebzadeh, F. Salboukh, and L. Deshmukh, “Forecasting Gold Prices with MLP Neural Networks: A Machine Learning Approach,” *International Journal of Science and Engineering Applications (IJSEA)*, vol. 13, no. 08, pp. 13–20, 2024.

[9] Wibawa, A.P., Utama, A.B.P., Lestari, W., Saputra, I.T., Izdihar, Z.N., Pujianto, U., Haviluddin, H. and Nafalski, A., “Mean-Median Smoothing Backpropagation Neural Network to Forecast Unique Visitors Time Series of Electronic Journal,” *Journal of Applied Data Sciences*, vol. 4, no. 3, pp. 163–174, 2023.

[10] F. Marini and B. Walczak, “Particle swarm optimization (PSO). A tutorial,” *Chemometrics and Intelligent Laboratory Systems*, vol. 149, no. part b, pp. 153–165, 2015.

[11] S. Mirjalili and S. Mirjalili, “Genetic algorithm,” *Evolutionary algorithms and neural networks: theory and applications*, vol. 780, no. 06, pp. 43–55, 2019.

[12] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey wolf optimizer,” *Advances in engineering software*, vol. 69, no. 03, pp. 46–61, 2014.

[13] G. Cacciotti, J. C. Hayton, J. R. Mitchell, and D. G. Allen, “Entrepreneurial fear of failure: Scale development and validation,” *J Bus Ventur*, vol. 35, no. 5, pp. 25, 2020.

[14] I. Muis, A. N. Hamid, and others, “Fear of failure and Entrepreneurial intentions in University Students.,” *Journal of Educational, Health and Community Psychology (JEHCP)*, vol. 13, no. 2, pp. 531, 2024.

[15] E. C. Santos, A. R. Galvão, C. S. Marques, and T. Mendes, “Enlightening the shadow dimensions of part-time entrepreneurship: Navigating fear of failure and enhancing social status,” *Strategic Change* vol. 33, no. 6, pp. 497-512, year 2024.

[16] M. Shahid Satar, G. Alarifi, A. A. Alkhoraif, and M. Asad, “Influence of perceptual and demographic factors on the likelihood of becoming social entrepreneurs in Saudi Arabia, Bahrain, and United Arab Emirates–an empirical analysis,” *Cogent Business and Management*, vol. 10, no. 3, pp. 2253577, 2023.

[17] C. Camelo-Ordaz, J. P. Diánez-González, and J. Ruiz-Navarro, “The influence of gender on entrepreneurial intention: The mediating role of perceptual factors: La influencia del género sobre la intención emprendedora: El papel mediador de los factores de percepción,” *BRQ business research quarterly*, vol. 19, no. 4, pp. 261–277, 2016.

[18] S. S. Sagar and J. Stoeber, “Perfectionism, fear of failure, and affective responses to success and failure: The central role of fear of experiencing shame and embarrassment,” *J Sport Exerc Psychol*, vol. 31, no. 5, pp. 602–627, 2009.

[19] C. K. Lee, G. W. Cottle, S. A. Simmons, and J. Wiklund, “Fear not, want not: Untangling the effects of social cost of failure on high-growth entrepreneurship,” *Small Business Economics*, vol. 57, no. 1, pp. 531–553, 2021.

[20] P. Chapman and R. A. Phillips, "Entrepreneurial fear of failure: An international comparison of antecedents and impact on venture creation," *Journal of the International Council for Small Business*, vol. 3, no. 4, pp. 281–291, 2022.

[21] J. C. Hayton, G. Cacciotti, A. Giazitzoglu, J. R. Mitchell, and C. Ainge, "Understanding fear of failure in entrepreneurship: A cognitive process framework," *Frontiers of Entrepreneurship Research*, vol. 33, no. 6, p. 1, 2013.

[22] S. Ansari, K. A. Alnajjar, S. Abdallah, M. Saad, and A. A. El-Moursy, "Parameter tuning of MLP, RBF, and ANFIS models using genetic algorithm in modeling and classification applications," in *2021 International Conference on Information Technology (ICIT)*, vol. 2021, no. 06, pp. 660–666, 2021.

[23] F. Itano, M. A. de A. de Sousa, and E. Del-Moral-Hernandez, "Extending MLP ANN hyper-parameters Optimization by using Genetic Algorithm," in *2018 International joint conference on neural networks (IJCNN)*, vol. 2018, no. 08, pp. 1–8, 2018.

[24] V. Rajalakshmi and S. G. Vaidyanathan, "MLP-PSO Framework with Dynamic Network Tuning for Traffic Flow Forecasting.," *Intelligent Automation amd Soft Computing*, vol. 33, no. 3, pp. 1335-1348, 2022.

[25] D. Sarkar, T. Khan, F. A. Talukdar, and S. R. Rengarajan, "Hyperparameters tuning of prior knowledge-driven multilayer perceptron model using particle swarm optimization for inverse modeling," in *2022 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting (AP-S/URSI)*, vol. 22, no. 09, pp. 441–442, 2022.

[26] Tang, C., Huang, C., Chen, Y., Heidari, A.A., Wang, S., Chen, H. and Zhang, Y., "Multi-strategy Grey Wolf Optimizer for Engineering Problems and Sewage Treatment Prediction," *Advanced Intelligent Systems*, vol. 6, no. 7, pp. 2300406, 2024.

[27] C. Tian, S. Ni, Z. Wang, and Y. Zhang, "Application of Grey Wolf Optimization Algorithm in Tuning Controller Parameters of Hypersonic Vehicle," in *2022 41st Chinese Control Conference (CCC)*, vol. 2022, no. 10, pp. 323–328, 2022.

[28] S.-H. Liao, P.-H. Chu, and P.-Y. Hsiao, "Data mining techniques and applications–A decade review from 2000 to 2011," *Expert Syst Appl*, vol. 39, no. 12, pp. 11303–11311, 2012.

[29] I. Anwar, P. Thoudam, M. Samroodh, M. Thoudam, and I. Saleem, "The dataset on fear of failure, entrepreneurship education, psychological and contextual predictors of entrepreneurial intention," *Front Psychol*, vol. 13, no. 08, pp. 5, 2022.

[30] D. Stathakis, "How many hidden layers and nodes?," *Int J Remote Sens*, vol. 30, no. 8, pp. 2133–2147, 2009.

[31] S. Sharma, S. Sharma, and A. Athaiya, "Activation functions in neural networks," *Towards Data Sci*, vol. 6, no. 12, pp. 310–316, 2017.

[32] Makhadmeh, S.N., Al-Betar, M.A., Doush, I.A., Awadallah, M.A., Kassaymeh, S., Mirjalili, S. and Zitar, R.A., "Recent advances in Grey Wolf Optimizer, its versions and applications," *IEEE Access*, vol. 12, no. 08, pp. 22991-23028, 2023.